

# 求极小碰集的蛛网算法\*

何士玉, 刘志刚, 徐建芳, 李文帆

(西南交通大学 电气工程学院, 成都 610031)

**摘要:** 为降低空间复杂度和减少搜索时间, 结合极小碰集的特点和生物学中蜘蛛捕食思想, 提出了一种搜索极小碰集的蛛网算法。该方法考虑集合之间的相关性, 并构造能在蛛网上寻路的访问蜘蛛用于寻找蛛网内集合的所有极小碰集。在该算法中, 所提出的访问蜘蛛生成和搜索策略能够降低空间复杂度和减少搜索时间。将此算法与其他的极小碰集算法进行比较, 实验结果表明, 该算法在保证得到所有极小碰集的前提下, 具有较低的空间复杂度和较高的时间效率。

**关键词:** 基于模型诊断; 极小碰集; 蛛网; 访问蜘蛛

**中图分类号:** TP301.6      **文献标志码:** A      **文章编号:** 1001-3695(2013)03-0684-04

**doi:**10.3969/j.issn.1001-3695.2013.03.010

## Cobweb algorithm of computing minimum hitting sets

HE Shi-yu, LIU Zhi-gang, XU Jian-fang, LI Wen-fan

(College of Electrical Engineering, Southwest Jiaotong University, Chengdu 610031, China)

**Abstract:** To reduce space complexity and reduce search time, this paper proposed a cobweb algorithm for search minimum hitting sets, which combined the characteristics of minimum hitting sets and thought on spider prey in biology. It considered this method the correlation between the sets, and structured the visiting spider which could find the path of the cobweb access to find all minimum hitting sets of sets within the cobweb. In the algorithm, the generation of visiting spiders and search strategy could reduce space complexity and reduce search time. Compared with the other algorithms which computing the minimum hitting sets, the results of experiments show that under the premise of guaranteeing to get all minimum hitting sets, the algorithm has lower space complexity and higher efficiency.

**Key words:** model-based diagnosis; minimum hitting set; cobweb; visiting spider

基于模型诊断是为了克服传统故障诊断方法的缺点而提出的一项新型智能推理技术<sup>[1]</sup>。其基本思想是利用系统模型, 根据系统预测值与实际观测结果的差异, 经过逻辑推导出故障。一般根据系统的描述及观测, 得到极小冲突集, 然后再由极小冲突集计算得到极小碰集, 即为系统的诊断。现实和理论中的很多问题在某种程度上都可以转换成极小碰集问题。例如, 基于模型诊断中极小诊断问题<sup>[2]</sup>、极小覆盖集问题、老师与课程 (teacher and courses) 问题和规则冲突检测算法中位向量的碰集问题<sup>[3]</sup>等。迄今为止, 已对计算碰集的方法进行了许多研究和改进, 如 HS-TREE<sup>[2]</sup>、HS-DAG<sup>[4]</sup>、HST-TREE<sup>[5]</sup>、BHS-TREE<sup>[6]</sup>、Boolean<sup>[7]</sup>、逻辑数组方法<sup>[8]</sup>和遗传算法<sup>[9-11]</sup>、DPSO<sup>[12]</sup>、DMDSE-Tree<sup>[13]</sup>等。这些方法主要缺点集中表现在: a) 采用树的结构, 空间复杂度高<sup>[3-6]</sup>; b) 因剪枝而可能丢失某些正确解<sup>[2,6]</sup>; c) 需要建立树或图, 数据结构及算法实现比较烦琐<sup>[2,4-6]</sup>; d) 一般先求出所有碰集, 最后还需化简才能得出所有极小的碰集<sup>[7-9]</sup>; e) 某些随机搜索算法采用近似计算, 不能保证得到所有极小碰集, 正确度不高<sup>[10-13]</sup>。本文针对以上缺点, 通过对人工蛛网相关知识的学习<sup>[14,15]</sup>, 结合搜索极小碰集的本质, 提出了一种搜索极小碰集的蛛网方法, 制定了相应的访问蜘蛛生成、死亡与搜索策略。该方法的主要优点如下:

a) 蜘蛛搜索过程简单, 不需要建树或图, 算法易于编程实现; b) 访问蜘蛛有选择地生成其搜索策略, 使空间复杂度大大降低, 其复杂度远远小于  $O(2^n)$ ; c) 蜘蛛死亡后得到的搜索辐即为极小碰集, 避免了计算集合是否为集合簇的极小碰集的过程, 提高了搜索效率; d) 有选择地生成访问蜘蛛, 避免了非极小碰集的出现, 并且不会像树那样因剪枝而丢失正确解; e) 蛛网算法所求得的集合 (辐) 一定是极小碰集, 并能够保证求得所有的极小碰集, 不用先存储所有碰集, 再经化简才能得出所有极小碰集, 节约了内存的占用, 明显提高了搜索效率。

### 1 预备知识

#### 1.1 基于模型诊断的相关概念和定理<sup>[4]</sup>

**定义 1** 一个系统定义为一个三元组  $(SD, COMPS, OBS)$ , 其中: a)  $SD$  是系统描述, 是一阶谓词公式的集合; b)  $COMPS$  为系统的组成部件集合, 是一个有限的常量集; c)  $OBS$  为一观测集, 一阶谓词公式的有限集。

**定义 2** 冲突集。系统  $(SD, OBS, COMP)$  的一个冲突集是元件集  $\{c_1, \dots, c_k\} \subseteq COMP$ , 它使得  $SD \cup OBS \cup \{\neg AB(c_1), \dots, \neg AB(c_k)\}$  是不可满足的。一个冲突集是最小冲突集当且

**收稿日期:** 2012-06-26; **修回日期:** 2012-09-05      **基金项目:** 国家自然科学基金资助项目 (U1134205, 51007074); 国家教育部新世纪优秀人才支持计划项目 (NECT-08-0825); 中央高校基本科研业务费专项资金资助项目 (SWJTU11CX141)

**作者简介:** 何士玉 (1986-), 男, 硕士研究生, 主要研究方向为基于模型的配电网故障诊断 (zjyi1986@163.com); 刘志刚 (1975-), 男, 教授, 博导, 主要研究方向为现代信号处理技术在电力系统信号处理中的应用、智能监控、现代轨道交通技术; 徐建芳 (1989-), 男, 硕士研究生, 主要研究方向为基于模型的电力系统故障诊断; 李文帆 (1987-), 男, 硕士研究生, 主要研究方向为电力系统信号处理、软件开发。

仅当它的任何一个真子集都不是冲突集。其中,一元谓词  $AB$  意味着 abnormal,  $AB(c_1)$  表示元件  $c_1$  异常,  $\neg AB(c_1)$  表示元件  $c_1$  正常。

**定义3 碰集。** 设  $C$  是一个冲突集合簇,  $C$  的碰集是一个集合  $H$ ,  $H$  满足两个条件:  $H \subseteq \bigcup_{S \in C} S$ ;  $\forall S \in C, H \cap S \neq \emptyset$ 。一个碰集是最小碰集当且仅当它的任何一个真子集都不是碰集。

**定理  $\Delta$**  是  $(SD, OBS, COMP)$  的一个极小诊断, 当且仅当  $\Delta$  是  $(SD, OBS, COMP)$  的最小冲突集的最小碰集。

## 1.2 蛛网介绍

在自然界中, 蜘蛛大多是靠结网捕食为生。绝大多数蜘蛛网都可以近似看做是一个中心对称的结构。它包括由中心点辐射向外的丝线称为辐 (spoke), 用来作为蛛网的框架; 称辐上的螺旋线为弦 (spiral)<sup>[15]</sup>。蜘蛛主要是通过感受蛛网的震动来判断是否有猎物, 以及猎物在网上的位置。蛛网中的每一条辐会把震动传导到中心点, 所以蜘蛛通常会守在中心点等待信息。但有时候蜘蛛会隐藏在其他位置, 这时蜘蛛会从中心点引出一根丝线并将一只腿放在线上, 以便随时掌握蛛网的信息。这根线是传导蛛网震动的通信信号线。蜘蛛通过这根传导线会确定哪根辐传导了震动信息, 并很快通过辐逼近猎物<sup>[16]</sup>。

算法描述前, 仿照蜘蛛捕食生物学, 为求取极小碰集进行以下相关定义。

**定义4 蛛网捕食。** 通过感受蛛网的震动来判断是否有猎物, 以及猎物在网上的位置进行捕食。

**定义5 蛛网结构。** 尝试将几个无序且无关联的几个集合转换为一个以逻辑中心节点为中心的有序互联类似蛛网的网络。

**定义6 旋。** 各集合元素之间用虚线连接定为弦, 有几个集合就有几个旋。

**定义7 辐。** 从最内旋到最外旋, 蜘蛛所经过的路径集合。

本文受自然界中蜘蛛网结构特征以及蜘蛛捕食特征的启发, 结合搜索最小碰集的本质, 提出了一种基于蛛网的极小碰集搜索算法。该算法的主要思路是: 将几个无序且无关联的几个集合转换成逻辑上的人工蜘蛛网络, 即蛛网结构。由于考虑了集合之间的相关性, 包括集合与集合、各集合元素以及本集合元素的关系 (作为蜘蛛是否进行搜索捕食的信息), 产生访问蜘蛛进行搜索, 使搜索过程变得简单且易于实现。

## 2 算法介绍

### 2.1 算法描述

如果初始给定的冲突集  $CS$  不是最小的, 则要把其变为最小冲突集 (minimal conflict set, MCS), 即将全部冲突集的超集去掉, 简化即将构造的初始蛛网。然后按各集合元素的多少进行排序。为形象表示说明, 各集合之间元素用虚线连接成圆, 每一个圆 (集合) 定义为蛛网的一个弦, 最里面的弦为弦 1 即  $X_1$ , 以此类推  $X_2, X_3, \dots, X_n$ 。最里面的弦为初始弦, 最外面的为边缘弦。初始弦里面包含几个元素就会产生几个初始访问蜘蛛。每一个初始蜘蛛按照一定的规则衍生后代蜘蛛以及按照制定的搜索策略进行搜索, 直到访问蜘蛛全部死亡为止, 所通过的路径称之为辐。搜索完毕后, 每一个初始蜘蛛都会形成自己的蜘蛛网络。因此, 搜索极小碰集的问题就转变为搜索所有辐的问题。下面定义蜘蛛的衍生规则及死亡规则:

同一访问蜘蛛对下一弦进行搜索访问时, 若生成两个或两

个以上的后代蜘蛛, 那么这样的后代蜘蛛就为姐妹蜘蛛, 姐妹蜘蛛访问的节点为姐妹节点。

**后代蜘蛛的衍生规则:** 从初始蜘蛛开始衍生后代蜘蛛进行访问搜索; 访问蜘蛛会跳过含有祖先蜘蛛及该蜘蛛已访问节点的弦, 搜索到未含有已搜索节点的弦时衍生后代访问蜘蛛。

**后代蜘蛛的死亡规则:** 若在某点的访问蜘蛛在余下的所有弦都没有衍生访问蜘蛛, 则此访问蜘蛛死亡; 若访问蜘蛛已访问完边缘弦, 则此访问蜘蛛死亡。

**访问蜘蛛的搜索规则:** 访问蜘蛛搜索出不含有祖先蜘蛛和该访问蜘蛛已访问节点的弦, 并访问该弦上的节点。若该弦上的节点与祖先蜘蛛的姐妹蜘蛛访问的节点 (即祖先节点的姐妹节点) 相同, 则不被访问; 若该弦上的节点与该访问蜘蛛的姐妹蜘蛛访问的节点 (即当前访问蜘蛛访问节点的姐妹节点) 相同, 则不被访问。

根据上述定义可知, 初始蜘蛛为所有后代蜘蛛的祖先蜘蛛, 后代蜘蛛也可能为祖先蜘蛛。同时, 在访问蜘蛛搜索过程中, 蛛网是动态变化的, 且不规则。算法步骤如下:

设有冲突集合  $CS = \{C_1, C_2, \dots, C_k, \dots, C_m\} (k = 1, \dots, m)$ 。

a) 集合处理。如果冲突集簇中含有超集, 先要进行超集删除处理。

b) 构造初始蛛网。计算各集合元素的个数, 按元素多少进行集合排序, 构造初始蛛网。

c) 搜索辐:

(a) 祖先蜘蛛按已定义的衍生规则衍生出后代蜘蛛, 后代蜘蛛按照已定义的访问规则访问剩余弦的节点。

(b) 后代蜘蛛又会成为祖先蜘蛛, 继续步骤 (a) 的操作后, 再继续 (b) 的操作, 直到访问蜘蛛全部死亡为止。

d) 把所有搜索辐存放到初始化为空集的精英集合中。

步骤 c) 可以直接得到全部的最小碰集, 避免了先存储所有碰集再经化简后得出所有极小的碰集的过程, 节约了内存的占用, 显著提高了搜索效率。

下面对该方法从设计思路、复杂度以及性能等方面进行简单说明:

a) 步骤 a) 的目的。极小碰集是由极小冲突集计算所得, 所以首先进行超集删除处理, 得到所给冲突集的极小冲突集, 降低计算复杂度。

b) 步骤 b) 的目的。对集合进行蛛网初始化, 一方面更加符合蛛网的实际结构; 另一方面, 该步骤是本文所设定的蜘蛛搜索策略的基础, 保证蜘蛛搜索策略的完备性, 即避免访问蜘蛛跳过含有有用信息的节点或弦, 造成不能搜索到全部极小碰集, 丢失完备性 (实例分析中具体说明)。

c) 步骤 c) 的目的。在步骤 b) 的基础上进行步骤 c) 的操作。该步骤避免了大量超集的出现, 同时未遗漏对没有造成超集的节点搜索, 避免了有用信息的丢失, 保证了搜索辐的完备性, 从而得出全部的极小碰集。

d) 复杂度和性能分析。因该方法是以集合之间的相关性为依据, 所以时间效率以及空间复杂度与极小冲突集的相关性有关。各集合之间毫无相关性时, 即最坏情况下, 空间复杂度为  $O(2^n)$ ; 一般情况下, 要远低于  $O(2^n)$ 。

以上部分给出了本文提出的求极小碰集的蛛网算法, 并进行了相应的理论分析。下面给出实例实验验证分析, 并进一步从设计思路、复杂度以及性能等方面作进一步说明。

### 2.2 实例分析

对于冲突集合簇  $CS = \{ \{2, 4, 5\}, \{1, 2, 3\}, \{1, 3, 5\}, \{2,$

4,6}, {2,4}, {2,3,5}, {1,6}}。由于访问蜘蛛在访问搜索的过程中蛛网呈现动态变化,不宜用图描述,下面只给出该集合簇搜索辐的蛛网图,如图1所示。

Cobweb 算法搜索过程如下:

a) 由于{2,4,5}和{2,4,6}是{2,4}的超集,所以删除,冲突集合簇CS简化为MCS = {{1,2,3}, {1,3,5}, {2,4}, {2,3,5}, {1,6}},简化搜索过程。

b) MCS中,集合元素的个数分别为3,3,2,3,2。利用元素的多少进行集合排序:{2,4}, {1,6}, {1,2,3}, {1,3,5}, {2,3,5},从前到后分别定义为弦X<sub>1</sub>、X<sub>2</sub>、X<sub>3</sub>、X<sub>4</sub>、X<sub>5</sub>。

c) 由于X<sub>1</sub>上有两个节点,自动生成两个初始蜘蛛S<sub>2</sub>和S<sub>4</sub>,分别访问X<sub>1</sub>上的节点2和4。

(a)S<sub>2</sub>搜索弦X<sub>2</sub>,生成后代蜘蛛S<sub>1</sub>和S<sub>6</sub>,分别访问X<sub>2</sub>上的节点1和6。

①由于S<sub>2</sub>和S<sub>1</sub>已访问节点2,1,所以蜘蛛S<sub>1</sub>跳过含有节点2或1的弦X<sub>3</sub>、X<sub>4</sub>、X<sub>5</sub>,然后蜘蛛死亡。搜索辐为{2,1}。

②由于S<sub>2</sub>和S<sub>6</sub>已访问节点2,6,所以S<sub>6</sub>生成后代蜘蛛S<sub>3</sub>和S<sub>5</sub>访问弦X<sub>4</sub>上的节点3和5,而不生成后代蜘蛛S<sub>1</sub>,因为S<sub>6</sub>的姐妹蜘蛛S<sub>1</sub>已访问1,然后蜘蛛死亡。搜索辐为{2,6,3}, {2,6,5}。

将搜索辐{2,1}, {2,6,3}和{2,6,5}存入精英集合。

(b)S<sub>4</sub>搜索弦X<sub>2</sub>,生成后代蜘蛛S<sub>1</sub>和S<sub>6</sub>,分别访问X<sub>2</sub>上的节点1和6。

①由于S<sub>4</sub>和S<sub>1</sub>已访问节点4和1,所以蜘蛛S<sub>1</sub>跳过含有节点4或1的弦X<sub>3</sub>、X<sub>4</sub>,搜索弦X<sub>5</sub>,生成后代蜘蛛S<sub>3</sub>和S<sub>5</sub>,不生成后代蜘蛛S<sub>1</sub>,因为S<sub>1</sub>为祖先蜘蛛S<sub>4</sub>的姐妹蜘蛛。S<sub>3</sub>和S<sub>5</sub>访问节点3和5,然后蜘蛛死亡。搜索辐为{4,1,3}, {4,1,5}。

②S<sub>4</sub>和S<sub>6</sub>访问节点4,6后,S<sub>6</sub>搜索弦X<sub>3</sub>。由于S<sub>4</sub>和S<sub>6</sub>的姐妹蜘蛛S<sub>2</sub>和S<sub>1</sub>访问了节点2和1,所以S<sub>6</sub>只生成后代蜘蛛S<sub>3</sub>,访问节点3。弦X<sub>4</sub>、X<sub>5</sub>上均含有节点3,所以蜘蛛死亡。搜索辐为{4,6,3}。将搜索辐{4,1,3}, {4,1,5}, {4,6,3}存入精英集合。

d)得到精英集合BMS = {{2,1}, {2,6,3}, {2,6,5}, {4,1,3}, {4,1,5}, {4,6,3}}。

用蛛网算法得到的极小碰集MHS是:{2,1}, {2,6,3}, {2,6,5}, {4,1,3}, {4,1,5}, {4,6,3},与文献结果[7]一致,验证了该算法的有效性。由此可见,该蛛网算法能够直接得到全部的最小碰集,避免了先存储所有碰集,最后化简得出所有极小碰集的过程。

通过步骤a)得到MCS = {{1,2,3}, {1,3,5}, {2,4}, {2,3,5}, {1,6}}后,假设不按步骤b)的方法进行排序,随便给出一种顺序:{1,2,3}, {1,3,5}, {2,4}, {2,3,5}, {1,6},然后按照步骤c)和d),得出极小碰集{4,6,3}, {4,1,5}, {2,6,5}比实际结果少三个极小碰集,丢失了完备性,所以步骤b)保证了该方法的完备性。这些步骤相辅相成、不可分割。

### 3 统计比较分析

为了验证该算法的时间复杂度,将本算法与DMDSE-Tree、BHS-Tree和Boolean进行比较。这里仿造文献[13]中的例子:假设集合簇中集合个数M与每个集合元素的个数N相等,即M=N,各集合分别为{1,2,...,N}, {2,3,...,N+1}, ..., {M, M+1, ..., N+M-1},其中每例运算测试10次取平均值作为最后结果。利用C++6.0编程,运行环境为Windows XP,AMD

Sempron™ Processor 2800+, 1.61 GHz的CPU, 1.00 GB的内存。运行结果如图2和表1所示。

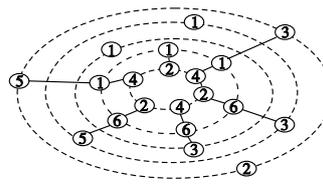


图1 搜索辐的蛛网图  
注: 为了便于描述,同一元素可在同一弦中重复出现。

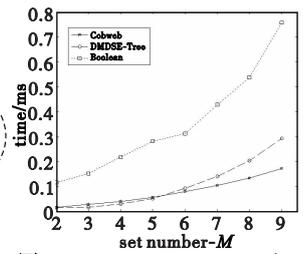


图2 Cobweb、DMDSE-Tree和Boolean方法运行时间对比  
注: 此图不再给出BHS-Tree的时间曲线,因它的效率明显低于其他三种方法。

表1 Cobweb、DMDSE-Tree、Boolean和BHS-Tree

方法运行时间对比

/ms

M	Cobweb	DMDSE	Boolean	BHS-Tree	MHS
2	0.018	0.015	0.120	0.421	2
3	0.029	0.020	0.152	0.602	4
4	0.041	0.031	0.221	0.847	7
5	0.057	0.055	0.280	1.319	11
6	0.080	0.098	0.310	2.014	16
7	0.105	0.150	0.435	3.322	22
8	0.135	0.210	0.570	5.218	29
9	0.172	0.295	0.765	8.710	37

从图2和表1可以看出:

a) 与DMDSE-Tree、BHS-Tree和Boolean相比,Cobweb算法的时间效率最好且具有绝对优势。

b) 虽说在M的值较小时,Cobweb算法的时间效率要低于动态极大度算法DMDSE-Tree,但随着M值的增大,Cobweb算法要优于DMDSE-Tree,且越来越好,在M=9时,效率约为DMDSE-Tree的两倍。Cobweb算法比DMDSE-Tree算法的计算时间增长平缓。因此,Cobweb算法比DMDSE-Tree算法更具有适用性。

c) Cobweb算法能够直接得出全部的极小碰集,避免了超集处理的过程。在保证正确性的前提下,具有较高的时间效率。

为验证该算法的空间复杂度,将本算法与HS-TREE进行比较,比较结果如表2所示。从表2可以看出:a)蛛网算法通过生成访问蜘蛛访问节点,访问的节点个数远小于HS-Tree,也就是说蛛网算法具有较低的空间复杂度;b) Cobweb算法访问节点少,不用存储大量的超集,大大降低了内存的占用。

表2 Cobweb、HS-TREE空间复杂度比较

M	访问节点数		搜索碰集数	
	Cobweb	HS-TREE	Cobweb	HS-TREE
2	4	5	2	3
3	7	13	4	9
4	11	33	7	25
5	16	81	11	65
6	22	193	16	161
7	29	449	22	385
8	37	1 025	29	897
9	46	2 305	37	2 049

综合上述实验结果,该算法中,访问蜘蛛有选择地生成及其搜索策略,不但使空间复杂度大大降低,其复杂度远远小于

$O(2^n)$ ,还简化了搜索过程,蜘蛛死亡后得到的搜索辐即为极小碰集,避免了计算集合是否为集合簇的极小碰集的过程,节约了计算时间。

#### 4 结束语

近年来国内外许多学者对极小碰集的求解方法进行了研究,主要集中在三个方面:a)基于树的数据结构求解;b)应用数学知识进行求解;c)近似求解。基于树的数据结构求解不易编程实现,且时间复杂度与空间复杂度较低。利用数学知识求解,如布尔代数 Boolean 方法,不需要建立树的结构,其空间复杂度和时间复杂度比基于树的方法都有较大降低。近似求解算法,如遗传算法和粒子群算法,以牺牲算法的完备性来换取时间效率,这类方法只能较快地计算出 95% 以上的碰集,而所有的极小碰集在很长的时间内都不能计算得到,不能保证其正确性。

本文提出的蛛网算法——Cobweb 算法,仿照生物学中的蜘蛛捕食,把集合元素之间的相关性作为蜘蛛进行搜索捕食的信息,有选择地生成访问蜘蛛,并按照捕食信息进行衍生和死亡,不但可以在较短的时间内得出全部正确解,且易于编程实现。由图 2 和表 1 可以看出,该方法要比空间复杂度和时间复杂度较好的 Boolean 算法好很多,更进一步说明了该算法的优越性。同时,不用像近似算法那样通过牺牲正确性来提高时间效率。换句话说,该算法在保证正确性的前提下,具有较高的时间效率和较低的空间复杂度,并节约了一定的内存占用。

本文给出的测试集合按照一定规律生成。下一步的工作可以探讨随机生成的集合之间的相关程度对空间以及时间复杂度的影响,同时在实际应用中检验该方法,建立复杂的基于模型诊断的配电网系统,利用该方法求取其极小碰集,以期在较短的时间内得出正确的故障诊断,满足实时性要求。

#### 参考文献:

[1] 欧阳丹彤,欧阳继红,刘大有. 基于模型诊断的研究与新进展[J].

吉林大学自然科学学报,2001(2):38-45.

- [2] REITER R. A theory of diagnosis from first principles[J]. *Artificial Intelligence*,1987,32(1):57-96.
- [3] 李林,卢显良. 一种基于位向量交集运算的规则冲突检测算法[J]. *计算机研究与发展*,2008,45(2):237-245.
- [4] GREINER R. A correction to the algorithm in Reiter's theory of diagnosis[J]. *Artificial Intelligence*,1989,41(1):79-88.
- [5] WOTAWA F. A variant of Reiter's hitting-set algorithm[J]. *Information Processing Letters*,2001,79(1):45-51.
- [6] 姜云飞,林笠. 用对分-HS 树计算最小碰集[J]. *软件学报*,2002,13(12):2267-2274.
- [7] 姜云飞,林笠. 用布尔代数求解最小碰集[J]. *计算机学报*,2003,26(8):920-924.
- [8] 林笠. 基于模型诊断中的用逻辑数组计算最小碰集[J]. *暨南大学学报*,2002,23(1):24-27.
- [9] 黄杰,陈琳,邹鹏. 一种求解极小诊断的遗传模拟退火算法[J]. *软件学报*,2004,15(9):1345-1350.
- [10] LIN Li, JIANG Yun-fei. Computing minimal hitting sets with genetic algorithm[C]//Proc of the 13th International Workshop on Principles of Diagnosis. 2002:77-80.
- [11] 张楠,孙吉贵,赵祥福,等. 求极小碰集的遗传算法[J]. *广西师范大学学报*,2006,24(4):62-65.
- [12] 蒋荣华,天书林,龙兵. 基于 DPSO 最小碰集算法的掩盖故障识别[J]. *系统工程与电子技术*,2009,31(4):997-1001.
- [13] 张立明,欧阳丹彤,曾海林. 基于动态极大度的极小碰集求解方法[J]. *计算机研究与发展*,2011,48(2):209-215.
- [14] 杨新宇,王鹏,曾明. MPLS 自治系统中基于人工蜘蛛的重路由研究[J]. *自然科学进展*,2004,14(7):85-90.
- [15] 蒋亚静,李远杰. 一种基于人工逻辑蛛网的路由算法[J]. *微机发展*,2004,14(11):21-24.
- [16] ZSCHOKKE S. Form and function of the orb-web[M]. Aarhus: Aarhus University Press,2000:99-106.

(上接第 683 页)

从表 1 中可以看出,在只引入粒子释放时,算法的性能并没有显著提高。这是因为 LDPSO 本身已经兼顾了收敛性和多样性,算法收敛速度并不快,出现收敛停滞的时间较晚,所以粒子释放策略作用有限。并且被释放后粒子的位置与个体最优、种群最优之间差距过大,导致粒子速度很大,难以再次快速收敛到种群最优附近,而粒子的释放速度又很快,造成算法的全局最优难以被再次更新。在只引入速度限制时,算法能通过前期的速度限制来加快粒子的收敛速度,但也容易出现因收敛速度过快而使多样性下降,导致算法过早收敛,影响算法的寻优精度。同时引入粒子释放和速度限制时,算法的收敛速度得到很好的保留,在进化后期出现收敛停滞时,算法也能通过粒子释放来增加多样性,并能很好地平衡粒子的释放速度和收敛速度,因而算法的寻优能力和收敛精度都有显著提高。

#### 参考文献:

- [1] KENNEDY J, EBERHART R C. Particle swarm optimization[C]//Proc of IEEE International Joint Conference on Neural Networks. 1995:1942-1948.
- [2] 林小朗,王磊. 改进粒子群优化算法的电力系统最优潮流计算[J]. *广东电力*,2007,20(3):12-15.
- [3] 唐雪琴,王侃,徐宗昌. 基于 MAPSO 算法的小波神经网络训练方

法研究[J]. *系统仿真学报*,2012,24(3):608-612.

- [4] SHI Yu-hui, EBERHART R. A modified particle swarm optimizer[C]//Proc of IEEE International Conference on Evolutionary Computation. 1998:69-73.
- [5] RATNAWEERA A, HALGAMUGE S, WATSON H. Particle swarm optimization with self-adaptive acceleration coefficients[C]//Proc of the 1st International Conference on Fuzzy Systems and Knowledge Discovery. 2003:246-268.
- [6] 田雨波,朱人杰,薛权祥. 粒子群优化算法中惯性权重的研究进展[J]. *计算机工程与应用*,2008,44(23):39-41.
- [7] 姚灿中,杨建梅. 一种基于有向动态网络拓扑的粒子群优化算法[J]. *计算机工程与应用*,2009,45(27):15-17.
- [8] 陈自郁,何中市,张程. 一种邻居动态调整的粒子群优化算法[J]. *模式识别与人工智能*,2010,23(4):586-592.
- [9] 王维博,冯全源. 基于分层多子群的混沌粒子群优化算法[J]. *控制与决策*,2010,25(11):1663-1668.
- [10] 孙湘,周大为,张希望. 一种混沌粒子群算法[J]. *计算机工程与科学*,2010,32(12):85-88.
- [11] 张玉芳,薛青松,熊忠阳. 基于禁忌搜索的动态粒子群算法[J]. *计算机工程与应用*,2008,44(24):56-58.
- [12] 方伟,孙俊,须文波. 基于微分进化算子的量子粒子群优化算法及应用[J]. *系统仿真学报*,2008,20(24):6740-6744.
- [13] 李季,孙秀霞,李士波,等. 基于遗传交叉因子的改进粒子群优化算法[J]. *计算机工程*,2008,34(2):181-183.